

CLAIMS:

1. A method of converting an explicit definition of a geometric object located in a prescribed space to a level set which implicitly defines said geometric object, the method comprising:

dividing the prescribed space into a grid of voxels, each voxel comprising a unit of the prescribed space;

casting rays in plural directions to obtain plural distance votes associated with each voxel in which each distance vote represents a distance from the voxel, through the prescribed space, to the geometric object in one of said plural directions of the rays;

assigning a magnitude value to each voxel by selecting the distance vote associated with that voxel which is lowest in magnitude;

determining if each voxel is either inside or outside of the geometric object in the prescribed space; and

assigning a sign value to each voxel, the sign value being either a positive sign or a negative sign in which each of the signs is representative of a sign condition of the voxel being either inside or outside of the geometric object.

2. The method according to Claim 1 wherein determining if each voxel is either inside or outside of the geometric object includes associating a sign vote with each distance vote associated with the voxel in which each sign vote is either a positive sign or a negative sign representative of the sign condition as determined by casting the rays, and wherein the sign value is assigned a sign which is a majority sign of the sign votes associated with that voxel.

3. The method according to Claim 2 wherein the explicit definition of the object includes normal vectors defined which project in a perpendicularly outward direction from a surface of the object, the method including determining the sign votes representing the voxel being either inside or outside of the object by comparison to the normal vectors of the surface of the object.

4. The method according to Claim 2 including either consistently assigning a positive sign or consistently assigning a negative sign to the sign value when sign votes of positive sign and sign votes of negative sign for a given voxel are equal in number.
5. The method according to Claim 1 including casting rays along at least three axes of each voxel which are lying transversely to one another.
6. The method according to Claim 1 including casting rays along axes of the grid of the voxels.
7. The method according to Claim 1 including casting rays in directions which intersect a plurality of voxels.
8. The method according to Claim 7 including interpolating distance votes of voxels between a selected voxel having a distance vote which has already been assigned and the object along an axis of the ray which was cast for assigning the distance vote to said selected voxel.
9. The method according to Claim 7 including interpolating distance votes of voxels between a selected voxel having a distance vote which has already been assigned and the object in a first direction along an axis of the ray which was cast for assigning the distance vote to said selected voxel and subsequently interpolating distance votes in a second direction along said axis by casting rays in an opposite direction.
10. The method according to Claim 1 including comparing the magnitude values of adjacent voxels adjacent to the implicitly defined object in the grid of voxels and adjusting the magnitude values such that adjacent ones of the magnitude values differ by approximately a magnitude of a dimension of the voxels and such that a surface of the implicitly defined object remains in a same position with the grid of voxels.
11. The method according to Claim 1 including applying a level set re-initialization algorithm to the voxel magnitude values which implicitly define the geometric object.

12. The method according to Claim 1 including defining a bounding box in the prescribed space which contains the object; casting rays only from voxels which are contained within the bounding box.

13. The method according to Claim 1 including defining a bounding box in the prescribed space which intersect the object and clipping the explicit definition of the object against the bounding box of the domain of the implicitly defined object.

14. The method according to Claim 1 wherein the explicit definition of the geometric object is a polygonal mesh.

15. The method according to Claim 1 wherein the explicit definition of the geometric object is a NURBS surface.

16. The method according to Claim 1 wherein the explicit definition of the geometric object is a subdivision surface.

17. The method according to Claim 1 including generating a representation of the explicit definition of the geometric object which allows for efficient ray casting prior to casting the rays.

18. The method according to Claim 1 including applying run length encoding to define a sparse level set in which consecutive voxels having identical values are compressed into a single value including a marker to indicate a quantity of the consecutive voxels.

19. The method according to Claim 18 including casting rays along a first axis using the run length encoding with the runs explicitly defined, and subsequently casting rays along second and third axes transverse to the first axis in which the values of the voxels are extracted from the run length encoding from the rays previously cast along a different one of the axes.

20. The method according to Claim 19 wherein the values of the voxels are extracted from the run length encoding from the rays previously cast using a one dimension Eikonal equation solved to estimate the magnitude values within predefined interface boundaries.

21. The method according to Claim 18 in which extra data structures are created and a different method using out-of-order traversal of the run length encoded signed distance data is performed to initialize the level set data in preparation for ray casting along a new axis by:

performing n^2 concurrent in-order traversals perpendicular to the axis along which a current line cross-section lies;

encoding the line cross-section using an additional run length encoded sparse line encoding by adding all the current perpendicular runs to a priority queue which orders them by the location on their axis of compression at which the runs terminate;

querying a priority queue to find out if there are any runs that change at this location in the previous axis of compression; and

sorting runs if said runs change their type from containing uncompressed data to compressed data so that one can linearly traverse the previous run length encoded sparse line encoding and encode the new run length encoded sparse line encoding by examining the runs which are changing.